2)

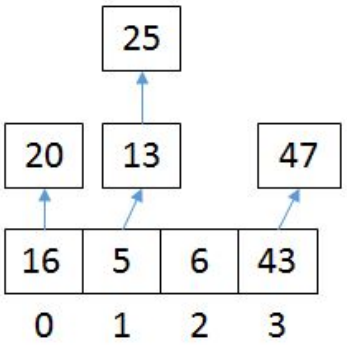
```
public static Arbol armarArbolDescendientes(Arbol a) {
    if (a != null) {
        Integer cantNodos = cantNodos(a) - 1;
        Arbol result = new Arbol(cantNodos);
        result.setIzq(armarArbolDescendientes(a.getIzq()));
        result.setDer(armarArbolDescendientes(a.getDer()));
        return result;
    }
    return null;
}

public static int cantNodos(Arbol a) {
    if (a == null)
        return 0;
    else
        return 1 + cantNodos(a.getIzq()) + cantNodos(a.getDer());
}
```

3) a)

```
public static void ordenar(Paquete[] paquetes) {
    // implementar cualquier algoritmo de ordenamiento
    // que ordene paquetes de cualquier manera dado su
    // tiempo de empaquetado
}

public static void main (String[] args)
{
    ArrayList<Paquete> paquetes = new ArrayList<Paquete>();
    ArrayList<Empleado> empleados = new ArrayList<Empleado>();

    ordenar(paquetes);

    int empleado_i = 0;
    int proximo = 1;
    for (Paquete p : paquetes) {
        empleados[empleado_i].asignar(p);
        empleado_i += proximo;

        if (empleado_i == 0)
            proximo = 1;
        else if (empleado_i == M-1)
            proximo = -1;
    }
}
```

4)

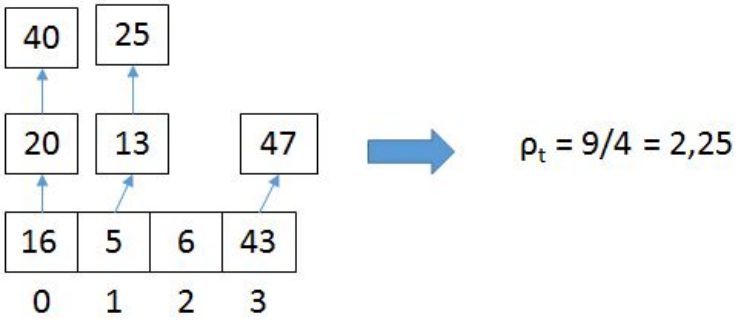
Condiciones iniciales:



ρ_d=2,3
F=0
M=4

- Inserción 40:

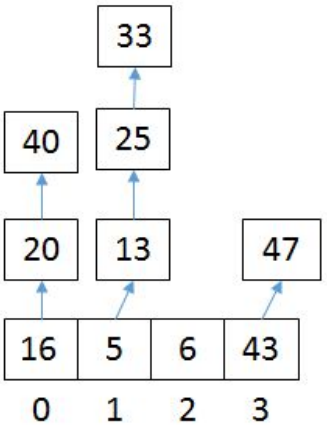
h(x) = 40 mod 4 = 0

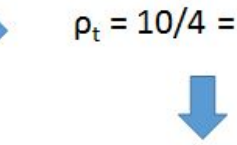


ρ_t = 9/4 = 2,25

- Inserción 33:

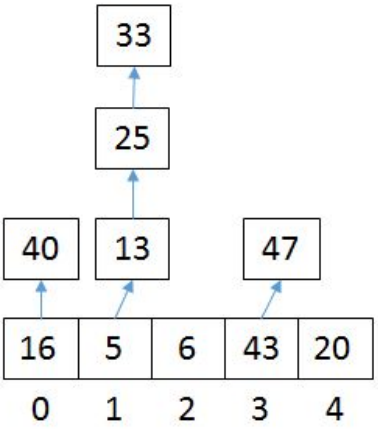
h(x) = 33 mod 4 = 1





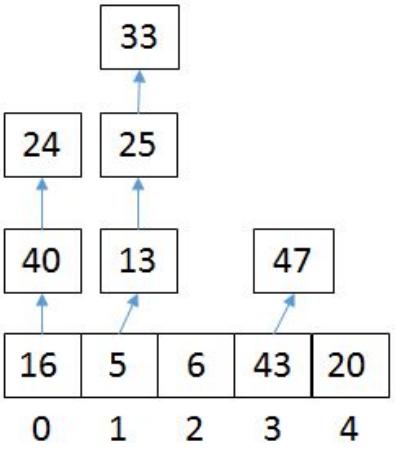
$\rho_t = 10/4 = 2,5$

- Corro frontera: F = 1
- Realeatorizo:
  - h'(16) = 16 mod 8 = 0
  - h'(20) = 20 mod 8 = 4
  - h'(40) = 40 mod 8 = 0



- Inserción 24:

h(x) = 24 mod 4 = 0



$\rho_t = 11/5 = 2,2$

5)

a) Falso. Las técnicas de hashing NO son eficientes para búsquedas por rango, ya que, al no haber orden, hay que consultar a TODOS los elementos por separado si pertenecen a determinado rango.
b) Falso. El algoritmo de ordenamiento Quicksort tiene una complejidad de O(n*log n)
c) Verdadero.
d) Falso. El lugar donde se insertan los elementos en una lista NO tiene impacto directo en el tiempo de búsqueda. Si la lista está desordenada, hay que recurrir a una búsqueda lineal.